

AUS920030436US1

Patent Application

IMAGE DISTRIBUTION IN DATA PROCESSING SYSTEMS

Inventors: Hung The Dinh

Mansoor A. Lakhdir

5

Phong Anh Pham

BACKGROUND OF THE INVENTION

10

Field of the Invention

The field of the invention is data processing, or, more specifically, methods, systems, and products for image distribution for documents in data processing systems.

15

Description Of Related Art

In distributing images on data processing systems, documents having many images are slow to load on the client side, and the images are cumbersome to administer on the server side. Consider the following HTML segment:

20

&lt;HTML&gt;

&lt;HEAD&gt;

&lt;TITLE&gt;Business Partner support from IBM PartnerWorld&lt;/TITLE&gt;

&lt;/HEAD&gt;

25

&lt;BODY&gt;

&lt;table width="760" border="0" cellspacing="0" cellpadding="0"&gt;

&lt;img src="//www.ibm.com/i/v11/m/en/mast\_logo.gif" border="0"

```
alt="IBM" width="150" height="47"/></td>
<td width="310" class="tbg"><a href="#main">
</a></td>
5 <table border="0" cellpadding="0" cellspacing="0">
<form name="Search" method="get" <input type="hidden" name="v"
value="11" size="15"/><tr>
</td>
10 <tr valign="middle"><td>
</td>
<td><input maxlength="100" class="input" size="15" name="q"
value="" type="text"/></td><td>
15 </td>
</BODY>
</HTML>
20
```

This example HTML segment is an excerpt from the IBM website at <http://www.developer.ibm.com>. Notice the repeated use of `<img>` elements. This segment contains five `<img>` elements, and the document from which this example was excerpted, at the time of this writing, contained 156 `<img>` elements. For each 25 such element, a browser displaying the document opens a separate TCP/IP connection to a server, and transmits an HTTP request message requesting the image file identified in the 'src' attribute of the `<img>` element. Each such request eventually

results in a corresponding HTTP response message from the server, through still another TCP/IP connection. In addition, each TCP/IP connection requires system calls to establish sockets and transmit TCP/IP ‘send’ messages, each of which requires a full-blown context switch at the CPU level, recognized by persons of ordinary skill in the art as a heavy computer processing burden. Displaying the document from which this example was excerpted requires 312 TCP/IP connections just for the image transfers. Moreover, this is not at all atypical. Web pages today often contain many images.

10 Notice also that the ‘src’ attributes identify image files in several file system locations. In fact, the ‘src’ attribute can only identify files stored in file system locations. System administrators on the server side must store and manage image files in ways that are cumbersome, with image files often scattered around in different file system locations on different servers. Tracking updates and locating and

15 removing obsolete images are all very cumbersome on file systems. For all these reasons, there is an ongoing need for improved ways of distributing images in data processing systems.

**SUMMARY OF THE INVENTION**

This specification discloses exemplary embodiments of methods, systems, and computer program products for distributing images in which advantageously all 5 images in a predefined group generally are returned at the same time, thereby greatly reducing the data communications connection burden of communicating images for display through markup documents. More particularly, exemplary embodiments of methods, systems, and computer program products are disclosed and explained for distributing images in a data processing system, including receiving in a client a 10 document for display, where the document contains markup according to a markup language, and the document also includes an image group identifier identifying a group of images; retrieving the images, from a server in the data processing system, in dependence upon the image group identifier; and displaying the images on the client according to the markup. In typical embodiments, retrieving the images is 15 carried out by retrieving all images in the group identified by the image group identifier before displaying any of the images.

Markup in documents typically includes a markup element that represents an instruction to retrieve, during a single communications connection to the server, all 20 images identified by the image group identifier, and the markup element comprises the image group identifier. Retrieving the images typically includes aggregating the images in a data structure on the client. The markup typically also includes markup elements that represent instructions to display images at display locations, and the markup elements comprise identifications of images in a data structure on the client.

25

Further embodiments of methods, systems, and computer program products are disclosed and explained for distributing images in a data processing system, including

storing images on a server; associating each image with at least one group of images identified by an image group identifier; receiving from a client a request for a group of images, the request comprising an image group identifier; retrieving from storage images identified by the image group identifier; and sending the retrieved images to the client. In some embodiments, storing images means storing images as BLOBs in a database, and associating each image with at least one group of images is carried out by storing an image identifier for each BLOB in association with an image group identifier for each file. In some embodiments, storing images means storing images as files on a file system, and associating each image with at least one group of images includes storing a pathname for each file in association with an image group identifier for each file.

Various embodiments include associating groups of images with an image retrieval routine so that retrieving images is carried out by invoking the image retrieval routine. Various embodiments include storing on a server documents comprising markup according to a markup language where each document includes at least one markup element containing an image group identifier identifying a group of images. In such embodiments, markup elements include identifications of individual images in a data structure on the client and represent instructions to display individual images at particular display locations.

The foregoing and other objects, features and advantages of the invention will be apparent from the following more particular descriptions of exemplary embodiments of the invention as illustrated in the accompanying drawings wherein like reference numbers generally represent like parts of exemplary embodiments of the invention.

**BRIEF DESCRIPTION OF THE DRAWINGS**

Figure 1 depicts an architecture for a data processing system in which various embodiments of the present invention may be implemented.

5

Figure 2 sets forth a block diagram of automated computing machinery.

Figure 3 sets forth a data flow diagram illustrating an exemplary method for distributing images in a data processing system.

10

Figure 4 sets forth a data flow diagram illustrating a further exemplary method for distributing images in a data processing system.

15

Figure 5 sets forth a database relationship diagram illustrating relations among records representing images and groups of images.

Figure 6 sets forth a data flow diagram illustrating a still further exemplary method for distributing images in a data processing system.

**DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS****Introduction**

5 Exemplary embodiments are described generally in this specification in terms of methods for image distribution in data processing systems. Persons skilled in the art, however, will recognize that any computer system that includes suitable programming means for operating in accordance with the disclosed methods also falls well within the scope of the present invention.

10

Suitable programming means include any means for directing a computer system to execute the steps of the method of the invention. Suitable programming means include, for example, systems comprised of processing units and arithmetic-logic circuits connected to computer memory. Such systems generally have the capability

15

of storing in computer memory programmed steps of methods according to exemplary embodiments for execution by a processing unit. Generally in such systems, computer memory is implemented in many ways as will occur to those of skill in the art, including magnetic media, optical media, and electronic circuits configured to store data and program instructions.

20

Further, embodiments may be implemented as a computer program product for use with any suitable data processing system. Embodiments of a computer program product may be implemented as a diskette, CD ROM, EEPROM ('flash') card, or other magnetic or optical recording media for storage of machine-readable

25

information as will occur to those of skill in that art. Persons skilled in the art will immediately recognize that any computer system having suitable programming means will be capable of executing the steps of methods according to exemplary

embodiments as included in a computer program product. Moreover, persons skilled in the art will recognize immediately that, although many of the exemplary embodiments described in this specification are oriented to software installed on computer hardware, nevertheless, alternative embodiments implemented as firmware or other computing machinery are well within the scope of the present invention.

Definitions

“BLOB” stands for “Binary Large OBject,” a collection of binary data stored as a single entity in a database. BLOBs are used to hold multimedia content such as video and audio clips, although they are also used to store software, even executable binary code. Images are typically expressed in binary encodings such as JPEG and GIF, and BLOBs are useful for storing images according to various embodiments of the present invention. Not all databases support BLOBs.

15

“Browser” means a web browser, which is a communications application for locating and displaying web pages. Browsers typically include a markup language interpreter, web page display routines, and an HTTP communications client. Typical browsers can display text, graphic images, audio and video. Browsers are operative in network-enabled devices, including wireless network-enabled devices such as network-enabled PDAs and mobile telephones. Browsers in wireless network-enabled devices often are downsized browsers called “microbrowsers.” Microbrowsers in wireless network-enabled devices often support markup languages other than HTML, including for example, WML, the Wireless Markup Language.

25

“CGI” means “Common Gateway Interface,” a standard technology for data communications of resources between web servers and web clients. CGI provides a

standard interface between servers and server-side ‘gateway’ programs that administer actual reads and writes of data to and from files systems and databases.

“Client,” “client device,” “client machine,” or “client computer” means any computer 5 or process requesting a service of another computer system or process using a protocol. Clients include, for example, personal computers, mainframes, PDAs, mobile telephones, laptop computers, devices capable of wireless as well as wireline communications, and any instrument capable of administering search queries and search results or responses. Clients may further include communications software 10 applications that establish connections for data communications with servers and issue requests for documents, images, and other resources.

A “communications application” is any data communications software capable of sending and receiving images distributed for documents or other data streams in data 15 processing systems. Examples include browsers, microbrowsers, special purpose data communications systems, server applications, and others as will occur to those of skill in the art.

“CPU” means ‘central processing unit.’ The term ‘CPU’ as it is used in this 20 disclosure includes any form of computer processing unit, regardless whether single, multiple, central, peripheral, or remote, in any form of computing machinery, including client devices, servers, and so on.

“Data processing system” means one or more computers, peripheral equipment, and 25 software that performs data processing. Data processing system is synonymous with ‘computer system,’ ‘computing system,’ and ‘information processing system.’

A “data stream” is any resource on any data processing system whose contents are organized by markup. Data streams include, for example, static files in markup languages, such as static HTML files or static HDML files. Data streams also include dynamically-generated content such as query results and output from CGI scripts,

5 Java<sub>TM</sub> servlets, Active Server Pages (“ASPs”), Java Server Pages (“JSPs”), and other kinds of dynamically-generated content as will occur to those of skill in the art.

“GUI” means ‘graphical user interface.’

10 “HDML” stands for ‘Handheld Device Markup Language,’ a markup language used to format content for web-enabled mobile phones. HDML is proprietary to Openwave Systems, Inc., and can only be operated on phones that use Openwave browsers. Rather than WAP, HDML operates over Openwave’s Handheld Device Transport Protocol (“HDTP”).

15

“HTML” stands for ‘HyperText Markup Language,’ a standard markup language for displaying web pages on browsers.

20 “HTTP” stands for ‘HyperText Transport Protocol,’ a standard data communications protocol of the World Wide Web.

A “hyperlink,” also referred to as “link” or “web link,” is a reference to a resource name or network address that allows the named resource or network address to be accessed. More particularly in terms of the present invention, invoking a hyperlink

25 implements a request for access to a resource, generally a document. Often a hyperlink identifies a network address at which is stored a resource such as a web page or other document. Hyperlinks are often implemented as anchor elements in

markup in documents. As the term is used in this specification, however, hyperlinks include links effected through anchors as well as URIs invoked through ‘back’ buttons on browsers, which do not involve anchors. Hyperlinks include URIs typed into address fields on browsers and invoked by a ‘Go’ button, also not involving anchors. In addition, although there is a natural tendency to think of hyperlinks as retrieving web pages, their use is broader than that. In fact, hyperlinks access “resources” generally available through hyperlinks including not only web pages but many other kinds of data as well as dynamically-generated server-side output from Java servlets, CGI scripts, and other resources as will occur to those of skill in the art.

10

An “image” is an electronic representation of a picture produced by means of sensing light, sound, electron radiation, or other emanations foment from the picture or reflected by the picture. An image also can be generated directly by software without reference to an existing picture. Images include pictures of scenes as well as pictures of graphical elements for display on computer screens. Images typically are expressed in digital formats, such as, for example, JPEG, GIF, PNG, TIFF, BIFF, bmp, Clear, FITS, IFF, NFF, OFF, PCX, TGA, and XBM. “JPEG” abbreviates “Joint Photographic Experts Group,” the original name of the committee that wrote the standard. “GIF” stands for “Graphics Interchange Format,” a format whose compression algorithm is proprietary to Unisys. “PNG” stands for “Portable Network Graphics,” a format developed as a non-proprietary alternative to GIF.

“The Internet” is a global network connecting millions of computers utilizing various protocols, including the Internet Protocol or ‘IP’ as the network layer of their networking protocol stacks. The Internet is decentralized by design, an example of a data processing system. An “internet” (uncapitalized) is any set of networks interconnected with routers.

“LAN” is an abbreviation for “local area network.” A LAN is a computer network that spans a relatively small area.

- 5     “Markup” means information added to a document to enable a person or system to process it. Markup is composed of syntactically delimited characters added to the data of a document to represent its structure. Markup information can describe the document’s characteristics, or it can specify the actual processing to be performed. Markup is composed of markup “elements,” each of which is defined by one or more tags. Markup elements may be defined with one or more “attributes.” Each attributes has a name and a value. The well known HTML anchor element, for example, includes a start tag `<a>` and an end tag `</a>`. The anchor element also attributes including, for example, an HREF attribute that is used to identify a URI for a hyperlink and a NAME attribute that is used to make an anchor available as a
- 10    hyperlink. An example of an anchor element is:
- 15

```
<a href="http://www.SomeWebSite.com/index.html"> Home </a>
```

- 20    This example establishes the word “Home” as an anchor of a hyperlink to the index.html document located at the URI identified by the HREF element, “http://www.SomeWebSite.com/index.html”. This example:

```

```

- 25    is an HTML image element `<img>`. HTML image elements reference images with hyperlinks identified by URIs in their SRC attributes. This example has a SRC attribute with a URI of “//www.ibm.com/i/mast\_logo.gif.” In addition, this example

<img> element has attributes defining image width and height. The image element is an example of an “empty” element in that, rather than having both a start tag and an end tag, it is composed of only the single tag <img>.

- 5 A “markup language” is a language used to define information (markup) to be added to the content of a document as an aid to processing it. Examples of markup languages include HDML, HTML, WML, XML, and many others. Markup elements in some markup languages are predefined by a standard for the language, as is the case for HDML, HTML, and WML, for example. Markup elements in other markup
- 10 languages are user defined, which is the case generally for XML and for SGML (the Standard Generalized Markup Language), the language upon which XML is based.

“PDA” refers to a personal digital assistant, a handheld computer useful as a client according to embodiments of the present invention.

- 15
- “Resource” means any aggregation of information administered in data processing systems according to embodiments of the present invention. Network communications protocols, such as, for example, HTTP, generally transmit resources, not just files. A resource is an aggregation of information capable of being identified
- 20 by a URI or URL. In fact, the ‘R’ in ‘URI’ stands for ‘Resource.’ The most common kind of resource is a file, but resources include dynamically-generated query results, the output of CGI scripts, dynamic server pages, and so on. It may sometimes be useful to think of a resource as similar to a file, but more general in nature. Files as resources include web pages, graphic image files, video clip files, audio clip files,
- 25 files of data having any MIME type, and so on. As a practical matter, most HTTP resources, WAP resources, and the like are currently either files or server-side script

output. Server side script output includes output from CGI programs, Java servlets, Active Server Pages, Java Server Pages, and so on.

A “server” is a computer that provides shared services to other computers over a network. Examples of servers include file servers, printer servers, email servers, web servers, and so on. Servers include any computer or computing machinery on a network that manages resources, including documents, and responds to requests for access to such resources. A “web server” is a server that communicates with other computers through data communications application programs, such as browsers or microbrowsers, by means of hyperlinking protocols such as HTTP, WAP, or HDTP, for example, in order to manage and make available to networked computers documents, images, and other resources.

“SQL” stands for ‘Structured Query Language,’ a standardized query language for requesting information from a database. Although there is an ANSI standard for SQL, as a practical matter, most versions of SQL tend to include many extensions. This specification provides examples of database queries against semantics-based search indexes expressed as pseudocode SQL. Such examples are said to be ‘pseudocode’ because they are not cast in any particular version of SQL and also because they are presented for purposes of explanation rather than as actual working models.

A “Java Servlet” is a program designed to be run from another program rather than directly from an operating system. “Servlets” in particular are designed to be run on servers from a conventional Java interface for servlets. Servlets are modules that extend request/response oriented servers, such as Java-enabled web servers. Java servlets are an alternative to CGI programs.

“TCP/IP” refers to two layers of a standard OSI data communications protocol stack.

The network layer is implemented with the Internet Protocol, hence the initials ‘IP.’

And the transport layer is implemented with the Transport Control Protocol, referred

5 to as ‘TCP.’ The two protocols are used together so frequently that they are often referred to as the TCP/IP suite, or, more simply, just ‘TCP/IP.’ TCP/IP is the standard data transport suite for the well-known world-wide network of computers called ‘the Internet.’

10 A “URI” or “Universal Resource Identifier” is an identifier of a named object in any namespace accessible through a network. URIs are functional for any access scheme, including for example, the File Transfer Protocol or “FTP,” Gopher, and the web. A URI as used in typical embodiments of the present invention usually includes an internet protocol address, or a domain name that resolves to an internet protocol

15 address, identifying a location where a resource, particularly a document, a web page, a CGI script, or a servlet, is located on a network, often the Internet. URIs directed to particular resources, such as particular documents, HTML files, CGI scripts, or servlets, typically include a path name or file name locating and identifying a particular resource in a file system connected through a server to a network. To the

20 extent that a particular resource, such as a CGI file, a servlet, or a dynamic web page, is executable, for example to store or retrieve data, a URI often includes query parameters, or data to be stored, in the form of data encoded into the URI. Such parameters or data to be stored are referred to as ‘URI encoded data,’ or sometime as ‘form data.’

25

“URI encoded data” or “form data” is data packaged in a URI for data communications, a useful method for communicating variable names and values in a

data processing system such as the Internet. Form data is typically communicated in hyperlinking protocols, such as, for example, HTTP which uses GET and POST functions to transmit URI encoded data. In this context, it is useful to remember that URIs do more than merely request file transfers. URIs identify resources on servers.

5    Such resources may be files having filenames, but the resources identified by URIs also may include, for example, queries to databases, including queries to search engines according to embodiments of the present invention. Results of such queries do not necessarily reside in files, but they are nevertheless data resources identified by URIs and identified by a search engine and query data that produce such

10    resources. An example of URI encoded data is:

`http://www.foo.com/cgi-bin/MyScript.cgi?field1=value1&field2=value2`

This example shows a URI bearing encoded data. The encoded data is the string  
15    "field1=value1&field2=value2." The encoding method is to string field names and field values separated by '&' and "=" with spaces represented by '+.' There are no quote marks or spaces in the string. Having no quote marks, spaces are encoded with '+,' and '&' is encoded with an escape character, in this example, "%26." For example, if an HTML form has a field called "name" set to "Lucy", and a field called  
20    "neighbors" set to "Fred & Ethel", the data string encoding the form would be:

`name=Lucy&neighbors=Fred+%26+Ethel`

"URLs" or "Universal Resource Locators" comprise a kind of subset of URIs, such  
25    that each URL resolves to a network address. That is, URIs and URLs are distinguished in that URIs identify named objects in namespaces, where the names may or may not resolve to addresses, while URLs do resolve to addresses. Although

standards today are written on the basis of URIs, it is still common to such see web-related identifiers, of the kind used to associate web data locations with network addresses for data communications, referred to as “URLs.” This specification uses the terms URI and URL more or less as synonyms.

5

“WAN” means ‘wide area network.’ One example of a WAN is the Internet.

“WAP” refers to the Wireless Application Protocol, a protocol for use with handheld wireless devices. Examples of wireless devices useful with WAP include mobile 10 phones, pagers, two-way radios, hand-held computers, and PDAs. WAP supports many wireless networks, and WAP is supported by many operating systems. WAP supports HTML, XML, and particularly WML (the Wireless Markup Language), which is a language particularly designed for small screen and one-hand navigation without a keyboard or mouse. Operating systems specifically engineered for 15 handheld devices include PalmOS, EPOC, Windows CE, FLEXOS, OS/9, and JavaOS. WAP devices that use displays and access the Internet run “microbrowsers.”

The microbrowsers use small file sizes that can accommodate the low memory constraints of handheld devices and the low-bandwidth constraints of wireless networks.

20

“WML” stands for ‘Wireless Markup Language,’ an XML language used as a markup language for web content intended for wireless web-enabled devices that implement WAP. There is a WAP forum that provides a DTD for WML. A DTD is an XML ‘Document Type Definition.’

25

“World Wide Web,” or more simply “the web,” refers to a system of internet protocol (“IP”) servers that support specially formatted, hyperlinking documents, documents

formatted in markup languages such as HTML, XML, WML, and HDML. The term “web” is used in this specification also to refer to any server or connected group or interconnected groups of servers that implement a hyperlinking protocol, such as HTTP, WAP, HDTDP, or others, in support of URIs and documents in markup 5 languages, regardless whether such servers or groups of servers are connected to the World Wide Web as such.

“XML” stands for ‘eXtensible Markup Language,’ a language that support user-defined markup including user-defined elements, tags, and attributes. XML’s 10 extensibility contrasts with most web-related markup languages, such as HTML, which are not extensible, but which instead use a standard defined set of elements, tags, and attributes. XML’s extensibility makes it a good foundation for defining other languages. WML, the Wireless Markup Language, for example, is a markup language based on XML. Modern browsers and other communications clients tend to 15 support markup languages other than HTML, including, for example, XML.

#### Data Processing Systems

Exemplary methods, system, and products for image distribution in data processing 20 systems are now explained with reference to the accompanying drawings, beginning with Figure 1. Figure 1 depicts an architecture for a data processing system in which various embodiments of the present invention may be implemented. The data processing system of Figure 1 includes a number of computers connected for data communications in networks. The data processing system of Figure 1 includes 25 networks (102, 104). Networks (102, 104) may be connected as LANs, WANs, intranets, internets, the Internet, webs, the World Wide Web itself, or other connections as will occur to those of skill in the art. Such networks are media that

may be used to provide data communications connections between various devices and computers connected together within the data processing system.

In Figure 1, servers 128 and 104 and storage unit 132 connect to network 102. In 5 addition, several exemplary client devices including a PDA 106, a workstation 108, and a mobile phone 110 are connected for data communications to network 102. Network-enabled mobile phone 110 connects to network 102 through wireless link 116, and PDA 106 connects to network 102 through wireless link 114. In the example of Figure 1, server 128 connects directly to client workstation 130 and 10 network 104 (which may be a LAN). Network 104 incorporates wireless communication links supporting a wireless connection to laptop computer 126. Network 104 also incorporates wireline protocols supporting a wired connection to client workstation 112.

15 The terms 'client' and 'server' are used generally to explain data communications according to the exemplary embodiments set forth in this disclosure. This use of the terms 'client' and 'server,' however, does not exclude peer to peer communications.

The particular servers and client devices illustrated in Figure 1 are for explanation, 20 not for limitation. Data processing systems useful according to various embodiments of the present invention may include additional servers, clients, routers, other devices, and peer-to-peer architectures, not shown in Figure 1, as will occur to those of skill in the art. Networks in such data processing systems may support many data communications protocols, such as, for example, TCP/IP, HTTP, WAP, HDTP, and 25 others as will occur to those of skill in the art. Various embodiments of the present invention may be implemented on a variety of hardware platforms in addition to those illustrated in Figure 1.

Figure 2 sets forth a block diagram of computing machinery that includes a computer 106, such as a client or server, useful in systems for image distribution in data processing systems according to embodiments of the present invention. The

5 computer 106 of Figure 2 includes at least one computer processor 156 or 'CPU' as well as random access memory 168 ("RAM"). Stored in RAM 168 is an application program 152. Application programs useful in implementing inventive methods of the present invention include servlets and CGI scripts running on servers and data communications programs such as browsers or microbrowsers running on client

10 machines. Also stored in RAM 168 is an operating system 154. Operating systems useful in computers according to embodiments of the present invention include AIX<sub>TM</sub>, Linux, Microsoft NT<sub>TM</sub>, and many others as will occur to those of skill in the art.

15 The computer 106 of Figure 2 includes computer memory 166 connected through a system bus 160 to the processor 156 and to other components of the computer. Computer memory 166 may be implemented as a hard disk drive 170, optical disk drive 172, electrically erasable programmable read-only memory space ('EEPROM' or 'Flash' memory) 174, RAM drives (not shown), or as any other kind of computer

20 memory as will occur to those of skill in the art.

The example computer 106 of Figure 2 includes communications adapter 167 implementing data communications connections 184 to other computers 182, servers, clients, or networks. Communications adapters implement the hardware level of data

25 communications connections through which client computers and servers send data communications directly to one another and through networks. Examples of communications adapters include modems for wired dial-up connections, Ethernet

(IEEE 802.3) adapters for wired LAN connections, and 802.11b adapters for wireless LAN connections.

The example computer of Figure 2 includes one or more input/output interface  
5 adapters 178. Input/output interface adapters in computers implement user-oriented input/output through, for example, software drivers and computer hardware for controlling output to display devices 180 such as computer display screens, as well as user input from user input devices 181 such as keyboards and mice.

10

#### Receiving and Displaying Images

Figure 3 sets forth a data flow diagram illustrating an exemplary method for distributing images in a data processing system that includes receiving (304) in a client (302) a data stream (e.g., document 306) for display, the document (306)

15 having markup (308) according to a markup language, the document further having an image group identifier (310) identifying a group of images (312). In the case of the web as a data processing system, receiving a document occurs because a user invokes a hyperlink through a data communications application such as a browser or microbrowser, and the application sends to a server identified in the hyperlink a  
20 request message according to a communication protocol supported on the web, such as HTTP, WAP, and so on, receiving in response the document identified in the hyperlink, which typically is a web page (static or dynamically generated on the server) expressed in a markup language such as HTML or WML.

25 In this example, the document includes an image group identifier (310) identifying a group of images (312). An image group identifier (310) may be included in a document as shown by the <IMGDB> element in the following exemplary markup:

```
<HTML>
  <HEAD>
    <TITLE>Business Partner support from IBM PartnerWorld</TITLE>
  5  </HEAD>
  <BODY>
    <IMGDB
      src="http://www.ibm.com/cgibin/retrieve.cgi?imageGroupID=myGroup"
      />
  10 <table width="760" border="0" cellspacing="0" cellpadding="0">
    <IMGID src="myGroup.mast_logo.gif" border="0" alt="IBM"
      width="150" height="47"/></td>
    <td width="310" class="tbg"><a href="#main">
      <IMGID src="myGroup.c.gif" border="0" width="1" height="1" alt="Skip
  15 to main content"/></a></td>
    <table border="0" cellpadding="0" cellspacing="0">
      <form name="Search" method="get" <input type="hidden" name="v"
        value="11" size="15"/><tr>
        <IMGID src="myGroup.printer.gif" width="23" height="19" alt="Link to
  20 printable version of page"/></td>
      <tr valign="middle"><td>
        <IMGID src="myGroup.pwhome.jpg" width="610" height="52" alt="IBM
          PartnerWorld home header graphic" /></td>
      <td><input maxlength="100" class="input" size="15" name="q" value=""
  25 type="text"/></td><td>
        <IMGID src="myGroup.fw.gif" width="16" height="16" alt="" /></td>
      </BODY>
```

</HTML>

This example is derived from the prior art example set forth above. This example, however, contains a new markup element: <IMGDB>. The new markup element

5 <IMGDB> represents an instruction to retrieve, during a single communications connection to the server, all images identified by an image group identifier, which is set forth in a 'src' attribute in the new markup element shown above. In this example, the image group identifier is URI encoded as "myGroup."

10 Because <IMGDB> is new, data communications applications such as browsers or microbrowsers may be adapted to respond to the new element by alterations at the source code level, by use of downloadable plug-ins such as Java applets, other client-side scripting means, or in other ways as will occur to those of skill in the art. The name of the new element <IMGDB> is an example only, not a limitation of the

15 invention. The new element can be named anything, <IMG-DOWNLOAD>, <GET-IMG>, <GET-GROUP>, and so on, as will occur to those of skill in the art, and all such names are well within the scope of the present invention.

The method of Figure 3 includes retrieving (316) the images, from a server (314) in

20 the data processing system, in dependence upon the image group identifier (310). In typical servers according to the present invention, images are stored in files systems or databases according to their respective image group identifiers. That is, each image is stored in association with an image group identifier so that all images in a group can be retrieved, gathered, and returned at approximately the same time, so

25 long as the server knows the desired image group identifier. Retrieving (316) images in dependence upon the image group identifier (310) therefore is typically carried out by transmitting from the client to a server identified in markup a request message

expressed in a data communications protocol and bearing an image group identifier.

In the example under discussion, the markup is:

```
5      <IMGDB
        src="http://www.ibm.com/cgi-bin/retrieve.cgi?imageGroupID=myGroup"
        />
```

The target server is identified by the domain name "www.ibm.com;" the protocol is identified as "http;" and the image group identifier is represented as "myGroup."

- 10 10 Retrieving (316) images in dependence upon the image group identifier (310) therefore in this example is carried out by transmitting from a client browser to the server at "www.ibm.com" an HTTP request message bearing the image group identifier URI encoded as "myGroup." In this example, the request message actually requests the resource identified as "/cgi-bin/retrieve.cgi," which is a CGI script called
- 15 15 by the server that takes a call parameter of an image group identifier, retrieves from a database the images associated with the image group identifier, and returns the images to the calling server, which in turn returns them to the requesting client in an HTTP response message.
- 20 20 The CGI script "retrieve.cgi" is server-side functionality for retrieving images. The use of a CGI script, however, is only for explanation, not a limitation of the invention.  
On a server supporting IBM's DB2 database management system, server-side functionality for retrieving images may be implemented as a DB2 'stored procedure.'
- 25 25 On a Java server, for a further example, server-side functionality for retrieving images can be implemented as a servlet and invoked with a URI such as, for example:  
src="http://www.ibm.com/servlets/retrieve?imageGroupID=myGroup"

Advantageously, all images in the group are returned at approximately the same time, that is, typically through a single request/response sequence. In fact, in many implementations of the method of Figure 3, retrieving (316) images (312) is carried out by retrieving all images in a group identified by an image group identifier (310) before displaying any of the images. Documents may include more than one markup element requiring an image group, that is, in the current example, more than one <IMGDB> element. In a document having only one <IMGDB> element, however, downloading all images in a group at the same time reduces the HTTP request/response traffic from one round trip per image to one round trip per document.

Typically according to the example of Figure 3, retrieving (316) the images includes aggregating the images in a data structure on the client. Aggregating images in a data structure on the client may be carried out, for example, by use of an array of structures in C, a two-element container object in C++, a hashtable in Java, and so on as will occur to those of skill in the art. An example of such an aggregation is a table named for an image group identifier, "myGroup," having two columns, one storing image identifiers and the other storing images:

20

myGroup	
Image IDs	Images
mast_logo.gif	---

c.gif	---
printer.gif	---
pwhome.jpg	---
fw.gif	---

This example, with a table named “myGroup” for an image group identifier and a column containing the image identifiers from the exemplary HTML segment above, supports references to the images of the form: groupImageID.imageID, so that a

5 reference to myGroup.mast\_logo.gif may be used by a browser or other communications application to retrieve the first image from the table. A reference to myGroup.c.gif returns the second image in the table, myGroup.printer.gif the third image, and so on. The images in this table are represented by dashes only because in this example the images are like BLOBs, the actual raw binary images themselves,

10 stored in computer memory client-side, ready for display through the browser.

In typical implementations of the method of Figure 3, the markup (308) includes markup elements that represent instructions to display images at display locations. The markup elements in this example include identifications of images in a data

15 structure on the client, and the method includes displaying (318) the images on the client (302) according to the markup (308). The HTML segment under discussion, for example, contains not only the new markup element <IMGDB>, but another new markup element as well: <IMGID>. The new element <IMGID> represents an instruction to display an image at a display location corresponding to the element’s

20 location in the logical structure of a document, that is, the structure provided by the markup itself, HTML tables, HTML paragraph marks, HTML line marks, HTML forms, and so on. The new element <IMGID> includes identifications of images in a

data structure on the client. In this example, the identifications of images in a data structure on the client take the form described above: groupImageID.imageID. More particularly, the example HTML segment sets forth five <IMGID> elements:

5                   <IMGID src="myGoup.mast\_logo.gif" border="0" alt="IBM" width="150" height="47"/>

10                   <IMGID src="myGoup.c.gif" border="0" width="1" height="1" alt="Skip to main content"/>

15                   <IMGID src="myGoup.printer.gif" width="23" height="19" alt="Link to printable version of page">

20                   <IMGID src="myGoup.pwhome.jpg" width="610" height="52" alt="IBM PartnerWorld home header graphic" />

25                   <IMGID src="myGoup.fw.gif" width="16" height="16" alt=""/>

The 'src' attributes in these examples, rather than identifying images located in remote file systems on remote servers across a data processing systems as was the case in prior art, now point to images in a data structure on the client machine itself, ready for quick retrieval and display. In the first example <IMGID> element, the attribute src="myGoup.mast\_logo.gif" identifies an image associated with the image identifier "mast\_logo.gif" in a data structure on the client named "myGroup." In the second example <IMGID> element, the attribute src=" myGoup.c.gif" identifies an image associated with the image identifier "c.gif" in a data structure on the client named "myGroup." In the third example <IMGID> element, the attribute

src="myGroup.printer.gif" identifies an image associated with the image identifier "printer.gif" in a data structure on the client named "myGroup." And so on.

#### Storing and Sending Images

5

Figure 4 sets forth a data flow diagram illustrating a further exemplary method for distributing images in a data processing system that includes storing 402 images 312 on a server 314 and, in the process of storing the images, associating 404 each image with at least one group of images identified by an image group identifier 412.

10

In some implementations of the method according to Figure 4, storing 402 images 312 is carried out by storing images as BLOBs in a database. In such implementations, associating 404 each image with at least one group of images is accomplished by storing an image identifier for each BLOB in association with an image group identifier for each file. In other implementations, storing 402 images is carried out by storing images as files on a file system. In such implementations, associating 404 each image with at least one group of images is accomplished by storing a pathname for each file in association with an image group identifier for each file. Either way, the availability of database management tools, automated queries, inserts, deletions, updates, reports, and so on, substantially eases the administrative burden on the server side and speeds response time with respect to methods of prior art.

25 Methods for storing (402) images (312) and associating (404) images with groups are further explained with reference to Figure 5. Figure 5 sets forth a database relationship diagram illustrating relations among records representing images and groups of images. In the table named "Image Groups" 512, each record represents a

group of images, and each record includes an image group identifier 412 and a text description field 514. In the table named “Images” 502, each record represents an image, and each record includes an image identifier 504, typically an identification code implemented as a string such as an image file name or an integer code, for 5 example. The Images table 502 in Figure 5 also includes images, stored in the table in the form of BLOBs 506. Moreover, the Images table 502 illustrates a further alternative form of image storage in which the images themselves are stored on a file system and the Image table 502 stores, for each image, a pathname specifying where in a file system the image is found in an image file.

10

A pathname is a sequence of subdirectory names that identifies a file. Every file in a file system typically has a name, called a filename, so the simplest type of pathname is just a filename. When a pathname is specified as only a filename, an operating system looks for that file in a current working directory. If the file resides in a 15 different directory, however, the operating system requires more information to find the file. The additional information is provided by specifying a path that the operating system must follow through a file system to find the file. The pathname always starts from a working directory or from a root directory. Each operating system has its own rules for specifying paths. In DOS, for example, the root 20 directory is named \, and subdirectories are separated in pathnames by additional backslashes. In UNIX, the root directory is named /, and subdirectories are separated in pathnames by additional slashes. In Macintosh operating systems, subdirectories in pathnames are separated by colons.

25 It is clear to readers of skill in the art that the Images table 502 and the Image Groups table 512 advantageously will have a many-to-many relationship because each image can be in many groups and each group may have many images in it. The many-to-

many relationship is advantageous because it helps to ‘normalize’ the database, avoiding storing the same image or the same image group definition more than once. The table named “Junctions” 510 in Figure 5 is a junction table which, in conjunction with the Images table 502 and the Image Group table 512 implements the many-to-many relationship. The Junctions table 510 contains one record for each combination of image and image group. That is, each record in the Junctions table represents one association of an image and an image group, and each record in the Junction table 510 contains an image identifier 504 and an image group identifier 412. The imageID field 504 in the Junctions table 510 is a foreign key to the Images table 502, forming a one-to-many relationship between the Images table 502 and the Junctions table 510. The groupID field 412 in the Junctions table 510 is a foreign key to the Image Group table 512, forming a one-to-many relationship between the Image Group table 512 and the Junctions table 510. The relationship of the Images table 502 and the Image Groups table 512, therefore, implemented through the Junctions table 510, is many-to-many 516.

The exemplary method of Figure 4 includes receiving (406) from a client (302) a request (414) for a group of images, the request comprising an image group identifier (310). Receiving a request comprising an image group identifier is typically carried out by receiving in a server, from a client browser, an HTTP request message bearing the image group identifier URI encoded as, for example: “imageGroupID=myGroup.” Such a request message typically originates in a client in response to markup like that illustrated earlier by:

25                   <IMGDB  
src=“<http://www.ibm.com/cgi-bin/retrieve.cgi?imageGroupID=myGroup>”  
/>>

In this example, as described above, the request message actually requests the resource identified as “/cgi-bin/retrieve.cgi,” a CGI script called by the server that takes a call parameter of an image group identifier (in this case “myGroup”), retrieves

5 from a database or a file system the images associated with the image group identifier, and returns the images to the calling server, which in turn returns them to the requesting client in an HTTP response message.

The method of Figure 4 further includes retrieving (408) from storage images (416)

10 identified by the image group identifier (310). In typical servers according to embodiments of the present invention, where images are stored as BLOBs in database records, retrieving images is carried out by parsing request messages into database queries, such as, for example:

15       SELECT imageBLOB FROM images, junctions  
          WHERE junction.groupID = “myGroup”  
          AND images.imageID = junctions.imageID;

This is an example SQL query having a form like the following:

20       SELECT imageBLOB FROM images, junctions  
          WHERE junction.groupID = “/\* insert image group identifier here \*/”  
          AND images.imageID = junctions.imageID;

Given a request message bearing URI encoded image group identifier “myGroup,”  
25 parsing the request message includes extracting the image group identifier from the request message and inserting it into the SQL query form. Asserting the example SQL query against a database of the form shown in Figure 5, extracts from the

database all the images in the image group designated as “myGroup.”

The method of Figure 4 also includes sending (410) the retrieved images to the client.

In this example, sending retrieved images is carried out by marshalling them into an

5     HTTP response message and transmitting the response message to the client that  
requested the images. Advantageously by use of this method, all the images for a  
document may be sent at the same time through a single data communications  
connection, such as, for example, a TCP/IP connection, representing a substantial  
efficiency in use of data communications resources and time as compared with prior  
10    art.

Figure 6 sets forth a data flow diagram illustrating a further exemplary method for  
distributing images in a data processing system. The method of Figure 6 includes  
storing (402) images (312), receiving (406) requests for images, retrieving (408)

15    images, sending (410) retrieved images, and so on, all as described above regarding  
the method of Figure 4.

The method of Figure 6 further includes associating (418) groups of images with an  
image retrieval routine (420), wherein retrieving the images is carried out by

20    invoking (422) the image retrieval routine. In typical embodiments, an image  
retrieval routine is server-side functionality for image retrieval, described above as  
being implemented as a CGI script, a Java servlet, or a DB2 stored procedure. In  
such embodiments, associating the groups with the routine is carried out by inserting  
the script, servlet, or stored procedure name in an HTML document in a markup  
25    element established for that purpose, such as, for example, the <IMGDB> element  
described in detail above. Invoking (422) the routine (420) then is carried out by  
calling the script, servlet, or stored procedure identified in the request message and

handing off the image group identifier as a call parameter to the retrieval routine.

The method of Figure 6 also includes storing (402) on a server (314) documents comprising markup according to a markup language, wherein each document further

- 5 includes at least one markup element containing an image group identifier identifying a group of images and markup elements that include identifications of individual images in a data structure on the client and represent instructions to display individual images at particular display locations. In typical embodiments, a markup element containing an image group identifier identifying a group of images is implemented as
- 10 described above with regard to the exemplary <IMGDB> element. In typical embodiments, markup elements that include identifications of individual images in a data structure on the client and represent instructions to display individual images at particular display locations are implemented as described above with regard to the exemplary <IMGID> element.

15

It will be understood from the foregoing description that modifications and changes may be made in various embodiments of the present invention without departing from its true spirit. The descriptions in this specification are for purposes of illustration only and are not to be construed in a limiting sense. The scope of the present

- 20 invention is limited only by the language of the following claims.